UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,006	10/23/2003	Robert G. Noble	1086.2013-001	8511

21005 7590 07/24/2007
HAMILTON, BROOK, SMITH & REYNOLDS, P.C.
530 VIRGINIA ROAD
P.O. BOX 9133
CONCORD, MA 01742-9133

EXAMINER	
CHEN, QING	
ART UNIT	**PAPER NUMBER**
2191	
MAIL DATE	**DELIVERY MODE**
07/24/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

5

Office Action Summary	**Application No.** 10/692,006	**Applicant(s)** NOBLE ET AL.	
	Examiner Qing Chen	**Art Unit** 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 June 2007*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 June 2007* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. This Office action is in response to the amendment filed on June 25, 2007.

2. **Claims 1-20** are pending.

3. **Claims 1-15** have been amended.

4. **Claims 16-20** have been added.

5. The objection to the oath/declaration is withdrawn in view of Applicant's submission of a supplemental oath/declaration.

6. The objection to the drawings is withdrawn in view of Applicant's amendments to the drawings.

7. The objections to the specification are withdrawn in view of Applicant's amendments to the specification.

8. The objections to Claims 1-15 are withdrawn in view of Applicant's amendments to the claims.

9. The 35 U.S.C. § 112, second paragraph, rejections of Claims 1-7, 10, and 12-14 are withdrawn in view of Applicant's amendments to the claims.

10. The 35 U.S.C. § 101 rejections of Claims 1-14 due to lack of tangible results are withdrawn in view of Applicant's arguments and amendments to the claims. The 35 U.S.C. § 101 rejection of Claim 15 due to electrical signals is withdrawn in view of Applicant's amendments to the claim. However, the 35 U.S.C. § 101 rejections of Claims 1-7 due to functional descriptive material are maintained in view of Applicant's arguments and amendments to the claims and further explained below.

Response to Amendment

Claim Rejections - 35 USC § 112

11. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

12. **Claim 20** is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 20 recites the limitation "the given software application domain terminology." There is insufficient antecedent basis for this limitation in the claim. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading "a given software application domain terminology" for the purpose of further examination.

Claim Rejections - 35 USC § 101

13. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

14. **Claims 1-7** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 1-7 are directed to computer apparatus. However, the recited components of the computer apparatus appear to lack the necessary physical components (hardware) to constitute a machine or manufacture under § 101. Although the claims recite a data server as a claimed element, the originally-filed specification disclose that such data server may be software or a mixture of hardware and software (emphasis added) *(see Page 14: 12-14)*. Therefore, these claim limitations can be reasonably interpreted as computer program modules—software *per se*. The claims are directed to functional descriptive material *per se*, and hence non-statutory.

The claims constitute computer programs representing computer listings *per se*. Such descriptions or expressions of the programs are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer, which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element, which defines structural and functional interrelationships between the computer program and the rest of the computer, that permits the computer program's functionality to be realized, and is thus statutory. See *Lowry*, 32 F.3d at 1583-84, 32 USPQ2d at 1035.

Claim Rejections - 35 USC § 102

15. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

16. **Claims 1-6, 8-13, and 15-19** are rejected under 35 U.S.C. 102(e) as being anticipated by **Little et al. (US 7,047,518)**.

As per **Claim 1**, Little et al. disclose:

- an editor defining (i) class views and (ii) a composite class view of the defined class views, given one or more software applications of interest and each given software application having a respective data model or data view, for each said given software application, the editor providing a class view of the respective data model *(see Figure 1: 106; Column 16: 65-67 through Column 17: 1-4, "Data entity groups represent the logical data model for the application. Each data entity group contains a set of relational table and relational view classes which represent RDBMS tables and views and any customized access definition." and 15-24, "To accommodate customized data access, the programmer can create custom access classes 240, illustrated in FIG. 16. In step 306, the programmer creates the relational table classes and a class diagram to show the relationship between them." and 25-30, "Composite view class deals with multiple table operation. The "AddCourseAndStudent" class will add one entry in the*

Course and one in the Student table. We use dependency to represent the relationship between a composite view class and the relational table classes involved. ");

- the editor consolidating said class views to form a composite class view *(see Column 17: 25-30, "Composite view class deals with multiple table operation. The "AddCourseAndStudent" class will add one entry in the Course and one in the Student table. We use dependency to represent the relationship between a composite view class and the relational table classes involved. ")*; and

- a data server instantiating a multi-tier data model, there being a core conceptual data model having a plurality of routes between attributes in the composite class view and attributes in the core conceptual data model, the class views effectively being one tier of the multi-tier data model, the composite class view effectively being a second tier of the multi-tier data model and the core conceptual data model effectively being a third tier, the multi-tier data model having links between corresponding attributes across tiers, the multi-tier data model providing management and sharing of engineering data of the given software applications with other process and plant engineering applications, and enhancing process engineering and plant operations *(see Figure 1: 104; Abstract, "... the plug-in works with Rational Software Corporation's Rational Rose modeling product and can be used to develop software applications for use with M3 and the Weblogic family of transaction and application server products from BEA Systems, Inc, and with other third-party software systems. "; Column 18: 5-25, "... a class diagram, a relational entity and an access class are created. The class diagram represents the actual pattern being applied to the relational entity. " and "The class diagram show the relationship of the generated classes and the classes coming from the framework. In general, this* *is the pattern that we use to handle the RDBMS-Object mapping for The Expert System. "; Column 23: 13-15, "The user may already have a good three-tier design defined and may need to just add the necessary details to make it a good M3 design. ").*

As per **Claim 2**, the rejection of **Claim 1** is incorporated; and Little et al. further disclose:

- an amalgamator that synthesizes the class views, the composite class view and the core conceptual data model into a consolidated multi-tier data model *(see Column 19: 16-26, "The very last step in the process is code generation. This is done by invoking "M3 Builder=>Generate ...". " and "All implementation details such as physical source files, make files, libraries are populated in all corresponding model components. ").*

As per **Claim 3**, the rejection of **Claim 1** is incorporated; and Little et al. further disclose:

- a mapper that links the core conceptual data model attributes to the composite class view and the composite class view attributes to class views, and provides a one-to-one mapping between an attribute in the composite class view and a route in the core conceptual data model to corresponding given software applications from which the attribute in the composite class view originated *(see Column 18: 21-25, "Each generated M3 Builder data entity group contains a set of data entity packages which map to the relational table and relational view classes. ").*

As per **Claim 4**, the rejection of **Claim 3** is incorporated; and Little et al. further disclose:

- wherein each class view is represented in terms from the respective given software application, and said given software application is able to access data from the core conceptual data model *(see Column 19: 37-49, "... the programmer can extend the generated code in several different ways to provide very powerful and robust data access handling: ... ").*

As per **Claim 5**, the rejection of **Claim 1** is incorporated; and Little et al. further disclose:

- wherein the class views, the composite class view and the core conceptual data model are represented by object oriented programming elements *(see Column 9: 26-29, "Source Code Generation: The Expert System applies certain mapping algorithms in the transformation of Rose logical and component views to generated Java and C++ server source code. ").*

As per **Claim 6**, the rejection of **Claim 5** is incorporated; and Little et al. further disclose:

- wherein certain object oriented programming elements are defined by classes *(see Column 10: 18-23, "M3 Framework is a logical package in the logical view, which contains all of the classes in the M3 Framework. Data Entity Framework is a logical package in the logical view, which contains a set of base classes from which all data entity classes must subclass from. ")*; and
- wherein the editor enables user creation and editing of definitions of classes *(see Column 10: 55-57, "An Edit Implementation feature 176 uses the a notepad program or other text editor to open the files associated with the components selected in the component diagram. ").*

As per **Claim 8**, Little et al. disclose:

- forming a multi-tier data model with links between corresponding attributes across tiers, a first tier being formed by:

- for each of multiple given software applications of interest and having a respective data model, providing a practitioner's view of the given software application using a respective class view of the respective data model *(see Column 16: 65-67 through Column 17: 1-4, "Data entity groups represent the logical data model for the application. Each data entity group contains a set of relational table and relational view classes which represent RDBMS tables and views and any customized access definition." and 15-24, "To accommodate customized data access, the programmer can create custom access classes 240, illustrated in FIG. 16. In step 306, the programmer creates the relational table classes and a class diagram to show the relationship between them.")*;

- a second tier being formed by consolidating class views into a composite class view *(see Column 17: 25-30, "Composite view class deals with multiple table operation. The "AddCourseAndStudent" class will add one entry in the Course and one in the Student table. We use dependency to represent the relationship between a composite view class and the relational table classes involved.")*; and

- a third tier being formed by forming a core conceptual data model having a plurality of routes between attributes in the composite class view and attributes in the core conceptual data model *(see Column 18: 5-25, "... a class diagram, a relational entity and an access class are created. The class diagram represents the actual pattern being applied to the relational entity." and "The class diagram show the relationship of the generated classes and the classes coming* *from the framework. In general, this is the pattern that we use to handle the RDBMS-Object mapping for The Expert System.")*; and

- sharing, via the multi-tier data model, engineering data of the given software applications with other process and plant engineering routines *(see Abstract, "... the plug-in works with Rational Software Corporation's Rational Rose modeling product and can be used to develop software applications for use with M3 and the Weblogic family of transaction and application server products from BEA Systems, Inc, and with other third-party software systems."; Column 19: 27-29, "Once the modeling is done, the programmer would generate 256 (shown in FIG. 23) the data access code for the application.").*

As per **Claim 9**, the rejection of **Claim 8** is incorporated; and Little et al. further disclose:

- wherein the second tier is formed by synthesizing the class views into the composite class view *(see Column 19: 16-26, "The very last step in the process is code generation. This is done by invoking "M3 Builder=>Generate ...". " and "All implementation details such as physical source files, make files, libraries are populated in all corresponding model components.").*

As per **Claim 10**, the rejection of **Claim 8** is incorporated; and Little et al. further disclose:

- wherein the step of forming a multi-tier data model further comprises producing a one-to-one mapping between an attribute in each class view to the composite class view, and a one-to-one mapping between an attribute in the composite class view and a route in the conceptual data model to corresponding given software applications from which the attribute in the composite class view originated *(see Column 18: 21-25, "Each generated M3 Builder data entity group contains a set of data entity packages which map to the relational table and relational view classes.").*

As per **Claim 11**, the rejection of **Claim 8** is incorporated; and Little et al. further disclose:

- wherein the step of providing a practitioner's view includes, in each class view, representing the respective data model in terms from the respective given software application *(see Column 16: 65-67 through Column 17: 1-4, "Data entity groups represent the logical data model for the application. Each data entity group contains a set of relational table and relational view classes which represent RDBMS tables and views and any customized access definition.").*

As per **Claim 12**, the rejection of **Claim 8** is incorporated; and Little et al. further disclose:

- representing at least one of the class views, the composite class view and the core conceptual data model in terms of object oriented programming elements *(see Column 9: 26-29, "Source Code Generation: The Expert System applies certain mapping algorithms in the transformation of Rose logical and component views to generated Java and C++ server source code.").*

As per **Claim 13**, the rejection of **Claim 12** is incorporated; and Little et al. further disclose:

- wherein certain object oriented programming elements are defined by classes *(see Column 10: 18-23, "M3 Framework is a logical package in the logical view, which contains all of the classes in the M3 Framework. Data Entity Framework is a logical package in the logical view, which contains a set of base classes from which all data entity classes must subclass from.")*; and

- enabling user creation and edition of definitions of classes *(see Column 10: 55-57, "An Edit Implementation feature 176 uses the a notepad program or other text editor to open the files associated with the components selected in the component diagram.").*

As per **Claim 15**, Little et al. disclose:

- a computer readable medium that manages engineering data *(see Column 33: 20, "A computer readable medium ...")*; and

- a set of computer program instructions encoded on the computer readable medium *(see Column 33: 20-22, "A computer readable medium, including instructions stored thereon ...")*, the set of computer program instructions when executed on a computer causing the computer to:

- provide a respective class view for each of plural given software applications of interest and having a respective data model, each class view being of the respective data model *(see Column 16: 65-67 through Column 17: 1-4, "Data entity groups represent the logical data model for the application. Each data entity group contains a set of relational table and relational* *view classes which represent RDBMS tables and views and any customized access definition."* *and 15-24, "To accommodate customized data access, the programmer can create custom access classes 240, illustrated in FIG. 16. In step 306, the programmer creates the relational table classes and a class diagram to show the relationship between them.")*;

- form a composite class view from the class views *(see Column 17: 25-30, "Composite view class deals with multiple table operation. The "AddCourseAndStudent" class will add one entry in the Course and one in the Student table. We use dependency to represent the relationship between a composite view class and the relational table classes involved.")*;

- form a conceptual model *(see Column 18: 5-25, "... a class diagram, a relational entity and an access class are created. The class diagram represents the actual pattern being applied to the relational entity." and "The class diagram show the relationship of the generated classes and the classes coming from the framework. In general, this is the pattern that we use to handle the RDBMS-Object mapping for The Expert System.")*;

- form a consolidated multi-tier data model from the class views, the composite class view and the conceptual model *(see Column 23: 13-15, "The user may already have a good three-tier design defined and may need to just add the necessary details to make it a good M3 design.")*; and

- via, the consolidated multi-tier data model, provide sharing of engineering data of the given software applications with other process and plant engineering applications *(see Abstract, "... the plug-in works with Rational Software Corporation's Rational Rose modeling product and can be used to develop software applications for use with M3 and the Weblogic family of transaction and application server products from BEA Systems, Inc, and with other third-party* *software systems."; Column 19: 27-29, "Once the modeling is done, the programmer would generate 256 (shown in FIG. 23) the data access code for the application.").*

As per **Claim 16**, the rejection of **Claim 15** is incorporated; and Little et al. further disclose:

- wherein the consolidated multi-tier data model insulates the given software applications from changes in the conceptual model *(see Column 19: 46-48, "... subclass from the generated classes and introduce the customized behavior in the subclass. This allows the generated class be refused in its original form.").*

As per **Claim 17**, the rejection of **Claim 15** is incorporated; and Little et al. further disclose:

- wherein the consolidated multi-tier data model is insulated from changes in the given software applications *(see Column 19: 46-48, "... subclass from the generated classes and introduce the customized behavior in the subclass. This allows the generated class be refused in its original form.").*

As per **Claim 18**, the rejection of **Claim 15** is incorporated; and Little et al. further disclose:

- wherein the consolidated multi-tier data model provides an application independent and normalized data model where the composite class view is application independent *(see Column 22: 2-13, "The Adapter pattern can be used in many situations: ... When you need to* *separate the client interface of an object from its implementation so each can evolve independently.").*

As per **Claim 19**, the rejection of **Claim 15** is incorporated; and Little et al. further disclose:

- wherein the consolidated multi-tier data model comprises an editor and a class store, the class store providing an interface to the respective class views, the composite class view, and the conceptual model to share data between the consolidated multi-tier data model and the given software applications *(see Figure 1: 106; Column 16: 55-63, "... create the Data Entity Framework category in the Rose model. This framework contains a set of base classes from which all data entity classes must subclass from.").*

Claim Rejections - 35 USC § 103

17. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

18. **Claims 7, 14, and 20** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Little et al. (US 7,047,518)**.

As per **Claim 7**, the rejection of **Claim 6** is incorporated; however, Little et al. do not disclose:

- wherein the editor employs an Extensible Markup Language.

Official Notice is taken that it is old and well known within the computing art to utilize XML. XML is widely used to facilitate the sharing of data across different information systems, particularly systems connected via the Internet. Formally defined languages based on XML (such as RSS, MathML, GraphML, XHTML, Scalable Vector Graphics, MusicXML and thousands of other examples) allow diverse software to reliably understand information formatted and passed in these languages. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the editor employs an Extensible Markup Language. The modification would be obvious because one of ordinary skill in the art would be motivated to exchange a wide variety of data on the Web and elsewhere.

As per **Claim 14**, the rejection of **Claim 13** is incorporated; however, Little et al. do not disclose:

- wherein the step of enabling user creation and edition includes employing Extensible Markup Language interfaces.

Official Notice is taken that it is old and well known within the computing art to utilize XML. XML is widely used to facilitate the sharing of data across different information systems, particularly systems connected via the Internet. Formally defined languages based on XML (such as RSS, MathML, GraphML, XHTML, Scalable Vector Graphics, MusicXML and thousands of other examples) allow diverse software to reliably understand information formatted and passed in these languages. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the step of enabling user creation and edition includes employing Extensible Markup Language interfaces. The modification would be obvious because one of ordinary skill in the art would be motivated to exchange a wide variety of data on the Web and elsewhere.

As per **Claim 20**, the rejection of **Claim 15** is incorporated; and Little et al. further disclose:

- wherein the composite class view provides for the class views remaining in a given software application domain terminology *(see Column 17: 25-30, "Composite view class deals with multiple table operation. The "AddCourseAndStudent" class will add one entry in the Course and one in the Student table. We use dependency to represent the relationship between a composite view class and the relational table classes involved.")*.

However, Little et al. do not disclose:

- wherein the editor and the class store use an Extensible Markup Language.

Official Notice is taken that it is old and well known within the computing art to utilize XML. XML is widely used to facilitate the sharing of data across different information systems, particularly systems connected via the Internet. Formally defined languages based on XML (such as RSS, MathML, GraphML, XHTML, Scalable Vector Graphics, MusicXML and thousands of other examples) allow diverse software to reliably understand information formatted and passed in these languages. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the editor and the class store use an Extensible

Markup Language. The modification would be obvious because one of ordinary skill in the art would be motivated to exchange a wide variety of data on the Web and elsewhere.

Response to Arguments

19. Applicant's arguments filed on June 25, 2007 have been fully considered, but they are not persuasive.

In the remarks, Applicant argues that:

a) Little does not discloses or suggest a computer apparatus that includes (i) a composite class view formed by a consolidation of class views, (ii) a core conceptual data model having a plurality of routes between attributes in the composite class view and attributes in the conceptual data model, and (iii) a multi-tier data model with the class views effectively being one tier, the composite class view effectively being a second tier and the core conceptual data model effectively being a third tier.

Applicants' synthesis or consolidation of the Class Views 20 results in the creation of the Composite Class View, which is an amalgamation and rationalization of the individual class views 20, and the Class Views remain in the application domain terminolog3,. See Applicants' specification at page 8, lines 1 through 4.

Contrast this with Little at step 308 of FIG. 14, and at Column 17, lines 25 through 36, which uses UML or a general purpose modeling language that includes a graphical notation that is used to create not a multi-tiered model, or any 'consolidated' composite class view, but instead an abstract model of the system, as shown in the multiple table operation at lines 23 through 36, or creates an "enhanced design UML model". See Little at Column 30, lines 66 through 67.

Contrast again, this graphical enhanced UML notation with Applicants' system where a data server 60 can instantiate data objects 12 and expose these objects through interfaces following the class and other views 20, 30, and 40 defined by class editor 55. This combined system (data server and class editor) enables the sharing of original application data with process plant engineering routines and programs.

Examiner's response:

a) Examiner disagrees. Little et al. clearly disclose:

- (i) a composite class view formed by a consolidation of class views *(see Column 17: 25-30, "Composite view class deals with multiple table operation. The "AddCourseAndStudent" class will add one entry in the Course and one in the Student table. We use dependency to represent the relationship between a composite view class and the relational table classes involved. ")*,

- (ii) a core conceptual data model having a plurality of routes between attributes in the composite class view and attributes in the conceptual data model *(see Column 18: 5-25, "... a class diagram, a relational entity and an access class are created. The class diagram represents the actual pattern being applied to the relational entity. " and "The class diagram show the relationship of the generated classes and the classes coming from the framework. In general, this is the pattern that we use to handle the RDBMS-Object mapping for The Expert System. ")*, and

- (iii) a multi-tier data model with the class views effectively being one tier, the composite class view effectively being a second tier and the core conceptual data model effectively being a third tier *(see Column 23: 13-15, "The user may already have a good three-tier design defined and may need to just add the necessary details to make it a good M3 design.").*

Furthermore, although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

In the remarks, Applicant argues that:

b) Moreover, Applicants rebut the contentions of the Office in that one would substitute portions of the UML model to use XML since, the whole point of Little's system is to use a freely available UML open source or freeware graphical modelling language that includes graphical notations to set relationships between groups of data to avoid a software engineer or software designer actually writing the software code. One of ordinary skill in the art would not attempt to edit in Little using XML interfaces or provide a class editing subsystem that employs an Extensible Markup Language for the UML graphical notations. The whole point of using an open source UML graphical system is to avoid drafting such code or employing such XML interfaces. These two concepts teach away from one another, which is a strong presumption in favor of patentability of Claims 7 and 14.

Moreover, Applicants are not aware of any systems that employ XML for model editors and ask the Office to provide such references to rebut the Office's contention that this is known.

Examiner's response:

b) As previously pointed out in the Non-Final Rejection (mailed on 04/04/2007) and currently maintained by the Examiner, Claims 7 and 14 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Little et al. The Examiner cites **Galea et al. (US 6,404,445)** as concrete evidence to support the Examiner's taking of Office Notice. The newly added reference is added only as directly corresponding evidence to support the prior common knowledge finding and it does not result in a new issue or constitute a new ground of rejection.

Galea et al. disclose:

- wherein the editor employs an Extensible Markup Language *(see Column 7: 27-30, "Modeler 304 is an XML editor which takes the user interface definition tags, merges the configuration domain and relation definitions, and spawns the execution of a compiler.")*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Galea et al. into the teaching of Little et al. to include wherein the editor employs an Extensible Markup Language. The modification would be obvious because one of ordinary skill in the art would be motivated to exchange a wide variety of data on the Web and elsewhere.

Conclusion

20. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

WEI ZHEN
SUPERVISORY PATENT EXAMINER

QC / QC
July 17, 2007